(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 993 178 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
12.04.2000 Bulletin 2000/15

(51) Int Cl.7: H04N 1/32

(21) Application number: 99307797.3

(22) Date of filing: 04.10.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 06.10.1998 JP 28411198
06.10.1998 JP 28411498

(71) Applicant: CANON KABUSHIKI KAISHA
Tokyo (JP)

(72) Inventors:
- Kakimoto, Atsushi, c/o Canon Kabushiki Kaisha
Tokyo (JP)
- Kuroshima, Masashi,
c/o Canon Kabushiki Kaisha
Tokyo (JP)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
High Holborn
2-5 Warwick Court
London WC1R 5DJ (GB)

(54) System for sharing image input device on network

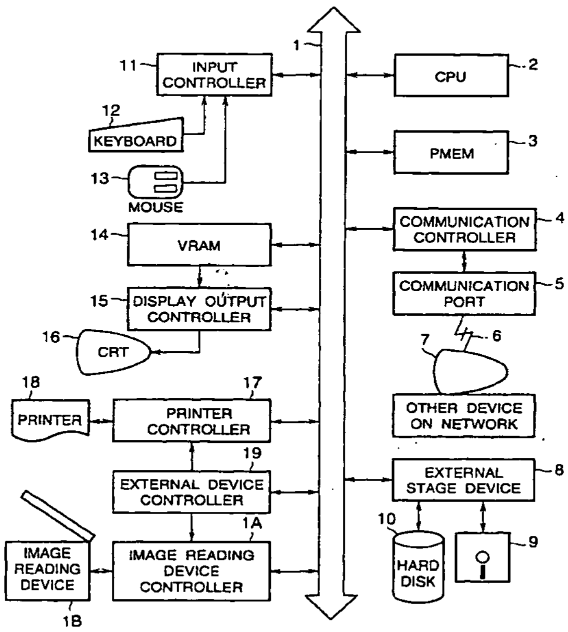
(57) It is an object of this invention to allow a user to perform operations while he or she is unconscious of the types of interfaces of various image input devices on a network.

An image input server device on a network is searched for, device control information of an image input device locally connected to the found image input server is obtained from the server, and virtual device information based on a universal image in-out interface is generated from the obtained device control information. An image reading session is executed in collaboration with the image input server device by using the virtual device information and information of communication with the image input server device.

An image input server device to which an image input device is locally connected receives a session request from a client on a network and determines the type of requesting client from the session request.

Different image reading sessions are executed when the determination result indicates that the requesting client is a dedicated client and that the requesting client is a universal client.

FIG.1



EP 0 993 178 A2

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to a system for sharing an image input device on a network and, more particularly, to a system capable of using an image input device regardless of the type of interface to the image input device.

Related Background Art

[0002] Conventionally, a system for sharing peripheral devices such as printers on a network such as a LAN has been realized. Such system has certain advantages: the cost can be reduced by sharing the peripheral devices on the network and the space for installing these devices can also be minimized.

[0003] In a system like this, however, it is impossible to share an image input device such as a scanner on a network and control the device by using a universal image in-out interface.

[0004] Since a universal in-out interface cannot control an image input device on a network, the user on the client device side who wishes to use the image input device must go to an apparatus to which the image input device is connected, directly operate on the apparatus, and store the obtained image data in a desired location.

[0005] Alternatively, the user starts a dedicated client application not corresponding to a universal image in-out interface to acquire an image via the network and opens another desired image editing application to edit the image data.

[0006] As described above, in either case the user must temporarily stop the work that he or she is currently executing in order to load an image from the network.

SUMMARY OF THE INVENTION

[0007] The present invention has been made to solve the above problems, and has as its object to share an image input device such as a scanner on a network and control the image input device by using a universal image in-out interface.

[0008] It is another object of the present invention to allow a user to perform operations while he or she is unconscious of the types of interfaces of various image input devices on a network.

[0009] It is still another object of the present invention to reduce the load on a network when a user uses an image input device on the network.

[0010] It is still another object of the present invention to properly extend functions of a client device using an image input device.

[0011] To achieve the above objects, an image input server device on a network is searched for, device control information of an image input device locally connected to the found image input server is obtained from the server, and virtual device information based on a universal image in-out interface is generated from the obtained device control information. An image reading session is executed in collaboration with the image input server device by using the virtual device information and information of communication with the image input server device.

[0012] An image input server device to which an image input device is locally connected receives a session request from a client on a network and determines the type of requesting client from the session request.

[0013] Different image reading sessions are executed when the determination result indicates that the requesting client is a dedicated client and that the requesting client is a universal client.

[0014] Other objects of the present invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] Fig. 1 is a block diagram showing the arrangement of an information processing apparatus in the first embodiment;

[0016] Fig. 2 is a view showing the arrangement of functions in an image input server device;

[0017] Fig. 3 is a view showing a setting screen of an image input device in a client device;

[0018] Fig. 4 is a view showing the data structure of image input device connection information;

[0019] Fig. 5 is a flow chart showing processing when a virtual image input device control module 39 is requested to access an image input device on a network;

[0020] Fig. 6 is a flow chart showing processing when the client device sends a request to an image input device connected to a server device;

[0021] Fig. 7 is a view showing an example of a virtual image reading device connection setting screen;

[0022] Fig. 8 is a view showing the arrangement of functions in a server extension module 22 in the second embodiment;

[0023] Fig. 9 is a flow chart showing processing when a client device sends a processing request to the server module 22;

[0024] Fig. 10 is a view showing the structure of session management information from a client to an image input device shared on a network in the second embodiment;

[0025] Fig. 11 is a view showing a display example when a client function extension module is incorporated into a universal client application in the second embodiment; and

[0026] Fig. 12 is a flow chart showing internal processing when the client extension module is incorporated into the universal client application.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0027] Embodiments of the present invention will be described in detail below.

[0028] Fig. 1 is a block diagram showing the arrangement of an information processing apparatus of one embodiment according to the present invention. A system bus 1 is connected to individual building blocks to be described below. A program memory (to be referred to as a PMEM hereinafter) 3 selects and loads a relevant program from a hard disk 15, and a CPU (Central Processing Unit) 2 executes the program. Input data from a keyboard 12 is stored as code information in the PMEM which is also a text memory. A communication controller 4 controls in-out data at a communication port 5. An output signal from the communication port 5 is transmitted to a communication port of an other device 7 on a network via a communication line 6. Data exchange with an imagq output device such as a printer and an image input device such as a scanner shared on the network is performed via the communication controller 4.

[0029] In this embodiment, a network such as a LAN will be described as this communication line 6. However, the present invention is, of course, applicable when the communication port and the communication channel connected to this communication controller are a general public port and channel.

[0030] The apparatus also includes an external storage device 8 and disks 9 and 10 for data files, e.g., a floppy disk (to be referred to as an FD hereinafter) 9 and a hard disk (to be referred to as an HD hereinafter) 10.

[0031] An input controller 11 is connected to input devices such as the keyboard 12 and a mouse 13. An operator operates the keyboard 12 and the mouse 13 to input operation commands to the system. In this embodiment, this mouse 13 is a pointing device (to be referred to as a PD hereinafter) for designating processing of image information on a CRT 16. An operator uses this mouse 13 to freely move the cursor on the CRT 16 in the X and Y directions to select a command icon on a command menu and designate processing and also designate an object to be edited and a drawing position.

[0032] The apparatus further includes a video image memory (to be referred to as a VRAM hereinafter) 14, a display output controller 15, and the CRT 16. Data displayed on the CRT 16 is mapped as bit map data on the VRAM.

[0033] A printer controller 17 controls data output from a printer 18 connected.

[0034] An image input device controller 1A controls an image input device 1B connected. These constituent elements 1A and 1B are essential components in an image input server device of this embodiment. However, they are not essential in a client device because, as described previously, the client device can use the same constituent elements shared by the server via the communication controller 4 and the communication port 5.

[0035] Furthermore, assume that the arrangement shown in Fig. 1 has the same functions regardless of whether the image input device and the image input device controller are physically discrete components or a single component contains both the image input device and the image input device controller.

<First Embodiment>

[0036] Fig. 2 shows the arrangement of functions in an image input server device of this embodiment.

[0037] Referring to Fig. 2, a network server module 21 converts data obtained by a server extension function (to be described later) into a data format exchangeable on a network corresponding to a universal network protocol.

[0038] Also, the network server module 21 converts network data transmitted from a client into an internal data format interpretable by a server extension module.

[0039] A server extension module 22 analyzes data transferred from the server, selects a designated one from a plurality of image service modules under the management of this module, and provides the service of the selected module. The server extension module 22 also transfers data from a certain service module to another service module (e.g., calls an image in-out control module from an OCR module).

[0040] When the server extension module requests the service of any image input device, a common interface conversion module 23 converts the request data into a control instruction corresponding to a designated input device control module 26. The common interface conversion module 23 also has a function of converting unique control information of a device from the input device control module into a general interface which can be transferred to the server extension module. More specifically, the former is image reading parameter setting information from a client, and the latter is image information and its attached information obtained as a result of reading designation.

[0041] An OCR module 24 is called when a client requests conversion from image data read by an image input device into document data. An image data format conversion module 25 converts the format of image data obtained from an image input device into an image data format designated by a client.

[0042] The input device control module 26 controls individual image input devices 27 in accordance with their respective properties. That is, the input device control module 26 inputs and outputs control instructions/information corresponding to (specialized to) the individual devices.

[0043] Fig. 3 shows the arrangement of functions in an image input client device of this embodiment.

[0044] Referring to Fig. 3, the apparatus requires an image input device connection information generating module 36.

[0045] Components 31 to 34 in Fig. 3 are necessary for a general application to access a locally connected image input device. A client extension function in this embodiment includes components 35 to 3A.

[0046] A general application 31 has an image input function. An image input device manager 32 provides this application with a universal image in-out interface. In accordance with designation from the in-out interface, this manager exchanges information with the designated one of image input device control modules under the management of the interface.

[0047] Image input device control modules 33 have a universal i/o. These modules control image input devices individually connected by the universal interface that is determined between these modules and the image input manager 32. Unlike the image input device control module 26 described above, this module provides a universal interface. That is, functionally each module is a combination of an image input device control module 26, a common interface conversion module 23, and a universal image in-out interface conversion module 37. The image input device manager 32 is, e.g., a TWAIN manager, and the image input device control modules 33 are equivalent to TWAIN drivers corresponding to the individual devices.

[0048] Individual image input devices 34 are locally connected to this apparatus.

[0049] Element items constructing client extension functions in the present invention will be described below.

[0050] A client function control module 35 controls all modules constructing individual client extension functions to be described below.

[0051] An image input device connection information generating module 36 acquires attribute information of an image input device connected to a server device, which is obtained using a network client module 38, and information necessary for network connection, and generates each image input device control information 3A.

[0052] At the same time, a virtual image input device control module 39 is automatically generated. When this module is generated, the image input device manager 32 can handle a device on the network as if it were virtually connected.

[0053] The universal image in-out interface conversion module 37 converts control information for an image input device on the server side, which is obtained using the network client module 36, into a universal interface corresponding to the image input manager 32.

[0054] This module provides the image input manager 32 with these interface functions not directly, but via individual virtual image input device control modules as will be described later.

[0055] The network client module 38 converts data into a data format exchangeable on a network corresponding to a universal network protocol so that the above modules and the server module can communicate with each other.

[0056] The virtual image input device control modules 39 are generated in one-to-one correspondence with set image input devices on the network. This module basically performs only intermediate processing; e.g., the module returns information of the corresponding image input device connection information 3A or transfers the information to the universal image in-out interface providing conversion module 37 and transfers the result to the image input manager 32.

[0057] As described above, the universal image in-out interface conversion module performs most of actual processing, so the module 39 can be a small module. Accordingly, the overhead can also be small even when these modules are generated for image input devices on a plurality of networks.

[0058] Fig. 4 shows the data structure of the image input device connection information shown in Fig. 3. Referring to Fig. 4, areas 401 to 404 indicate pieces of network connection information of the corresponding image input device.

[0059] The area 401 stores information for identifying a connecting destination server on the network, i.e., a machine name on the network or a network address.

[0060] The area 402 stores the network name of an image input device shared on the network, which is used to identify the device. The area 403 stores security authorization information for obtaining use authorization, when security limitations are set for the use of the device. This information is, e.g., a password for obtaining use authorization.

[0061] The area 404 stores other information necessary for network connection.

[0062] Areas 405 to 408 indicate pieces of identifying information of the device, which are used to provide virtual device information to the aforementioned image input device manager. These pieces of information are loaded from the server device and stored in these areas.

[0063] The area 405 stores name information of the device. The area 406 stores version information of a module. The area 407 stores information about the creator's name of the module. The area 408 stores other identifying information. Areas 409 to 414 indicate pieces of function information of the device. In this embodiment, n pieces of function information exist, and the areas 409 to 411 and the areas 412 to 414 respectively indicate sets of these pieces of function information.

[0064] The function information herein mentioned indicates the capability of the corresponding device, e.g., corresponding size information of a loading area or a list of loading resolutions.

[0065] The area 409 stores a function identifier. The area 410 stores a data format identifier. This information is used to identify whether the format of function information is a certain value, a value in a certain range, or a list form (1, 5, 10, 15).

[0066] The area 411 stores actual function information in accordance with the data format. The areas 412 to 414 store the nth function information in the same form.

[0067] Fig. 5 shows a flow chart of internal processing on the client side when the image input device manager sends, to the virtual image input device control module, a request of access to an image input device on the network. The procedure of this flow chart is realized by the CPU 2 by executing a program stored in the PMEM 3.

[0068] First, in step S51, the image input manager sends a request to the virtual image input device control module via the universal interface.

[0069] In step S52, the CPU 2 checks whether it is possible to process the request by the virtual image input device control module.

[0070] The virtual image input device control module can acquire static device control information such as device identifying information without inquiring of the image input device via the network. Therefore, the CPU 2 determines that it is possible to process the request.

[0071] To acquire image data, on the other hand, it is necessary to give an instruction to the image input device, so the CPU 2 determines that the request is impossible to process. This eliminates unnecessary use of the network and thereby reduces the load on the network.

[0072] If the CPU 2 determines in step S52 that it is possible to process the request, the flow branches to step S53; if not, the flow branches to step S54. In step 553, the CPU 2 generates data requested in this module by using the image input device connection information.

[0073] In step S54, the CPU 2 reads out necessary information from the image input device connection information and calls the universal image in-out interface conversion module.

[0074] In step S55, the CPU 2 checks whether it is possible to process the request by the universal image in-out interface conversion module.

[0075] That is, settings of, e.g., some attributes of the device are done only by settings in this module, and pieces of information set to this point are collectively transmitted to the server when it is actually necessary to access the network. This increases the efficiency.

[0076] If the CPU 2 determines in step S55 that it is possible to process the request in this module, the flow branches to step S56; if not, the flow branches to step S57.

[0077] In step S56, the CPU 2 processes the data required by the universal image in-out interface conversion module. In step S57, the CPU 2 sends a request from the image input device connection information to the server device on the network to which the corresponding image input device is connected.

[0078] If a response for the request sent to the server in the preceding step is received, in step S58 the CPU 2 converts the response into an understandable (universal) data format in the image input device manager by the universal image in-out interface conversion module.

[0079] By any of the above processes, the image input device manager can obtain desired return information.

[0080] Fig. 6 shows a flow chart of internal processing on the server side when the client device sends a request to an image input device connected to the server device.

[0081] First, in step S601 the CPU 2 receives a request from the client side via the network.

[0082] In step S602, the CPU 2 checks whether the request from this client is a session for the first time.

[0083] If the request is from a client that has already started a session, session id (identifying information) is added to the data obtained in step S601.

[0084] If the CPU 2 determines in step S602 that the request is the first session, the flow branches to step S603; if not, the flow branches to step S605.

[0085] In step S603, the CPU 2 checks whether this server device allows access to the requesting client. If access is allowed, the flow branches to step S604; if not, the CPU 2 sends access reject information to the client.

[0086] In step S604, the CPU 2 opens a session between the requesting client and the server and returns session id as a consequence. This session id is valid until the client side sends a close request or the server process in the server device is completed. The server can naturally establish sessions with a plurality of clients at the same time.

[0087] If the request is from a client with which a session is already open, in step S605 the CPU 2 checks whether the session id designated to the server is valid. If the CPU 2 determines in step S605 that the session id is valid, the flow branches to step S606; if not, the CPU 2 returns information indicating that the session is invalid.

[0088] In step S606, the CPU 2 checks whether the request from the client is a session close process. If YES in step S606, the flow branches to step S607; if NO in step S606, the flow branches to step S608.

[0089] In step S607, the CPU 2 executes the session close process with the designated client, and the flow proceeds on to a termination process.

[0090] In step S608, the CPU 2 checks whether the image input device designated by the client is valid. If YES in step S608, the flow branches to step S609; if NO in step S608, the CPU 2 returns information indicating that the device is an invalid printer to the client, and executes a termination process.

[0091] In step S609, the CPU 2 transfers the request information from the client to an image input device control module corresponding to this image input device. In step S610, the CPU 2 obtains return information from this image input device control module.

[0092] In step S611, the CPU 2 informs the client side of the result obtained in the preceding step via the network.

[0093] Fig. 7 shows an example of a virtual image reading device connection setting screen for realizing the functions of this embodiment.

[0094] By operation from the setting screen shown in

Fig. 7, virtual connection information is generated only for an image input device desired by the user.

[0095] Referring to Fig. 7, an area 71 is used to designate a network range, e.g., a domain, for searching for an image input device shared on the network. A list 72 shows image input devices found in this network range. In this embodiment, the sharing names of these image input devices are displayed in this list, and the network names of server devices to which these input devices are connected are described in parentheses. A list 73 shows image input devices for which virtual connection information is formed. The image input devices displayed in the list 73 are displayed in a device list as if they are image input devices directly connected to the client device. At the same time, actual image reading can be designated. A control button 74 is used to designate addition of virtual connection setting. When an operator selects a device from the list 72 and presses the button 74 by using a pointing device or the like, the selected device is added to the list 73. A control button 75 is used to designate deletion from the list. When deletion is designated, virtual connection information of the corresponding device is deleted. This makes the device unusable from the client device.

[0096] A control button 76 is used to designate image input device search in the designated network. A control button 77 is used to designate termination by reflecting the settings in this dialogue. A control button 78 is used to designate termination by canceling the settings.

<Second Embodiment>

[0097] In the second embodiment, an image input server device performs an image reading session by using an interface corresponding to the type of client.

[0098] Fig. 8 shows the arrangement of various functions in a server extension module 22.

[0099] A server extension function control module 81 controls all modules constructing the server extension module. This module 81 collectively interfaces with external devices, determines the type of requesting client, and calls a lower-order module having the corresponding function.

[0100] An interface supply module 82 for dedicated client supplies an image input device control interface to a dedicated client module corresponding to this server device.

[0101] An interface supply module 83 for universal client supplies an image input device control interface to a universal client application.

[0102] The difference between functions supplied by the interface supply module 82 for dedicated client and the interface supply module 83 for universal client is that the former, including an original function supplied by a server device in the present invention, can be used on the client module side, whereas the latter is limited to a function supplied as an interface interpretable by a universal application. As an example, when a universal application is an Internet browser, the interface supplied by the latter can exchange data by using a form function.

[0103] A universal client extension module exists to eliminate the function limitations on a universal client application as described above. This module can operate on a universal client application to extend the user interface and, as a consequence, allow detailed functions supplied by the server device to be used.

[0104] A download control module 84 for universal client extension module checks whether this universal client extension module exists on the client side or not, and downloads and incorporates the module if the universal client extension module does not exist.

[0105] As described above, when the universal client extension module is downloaded to the client side, the extended interface must be supplied. An interface supply module 85 for universal client extension module performs this processing.

[0106] A download objective client decision module 86 supplies, to a requesting client, a function of checking whether the client extension module can be downloaded. As a means for this check, it is possible to use information of a user who is currently logging on to the requesting client or a password requested when a user accesses the server.

[0107] When any of these high-order modules accesses a low-order image device control module, an interface control module 87 supplies a unified interface.

[0108] Fig. 9 shows a flow chart of an example of the flow of processing when a client device sends a processing request to the server extension module 22.

[0109] First, in step S91 whether the request from this client is the first image process session is checked. This is done by checking whether session id (to be described later) is added as request information from the client side.

[0110] If a session id is added, this session has been established previously. If the session is the first one, the flow branches to step S96; if not, the flow branches to step S92.

[0111] "Session" herein mentioned is information for holding request information when a specific client requests a specific shared image input device on the server side.

[0112] In step S92, whether this session request is from a dedicated client module is checked.

[0113] This can be done by checking client identifying information present in the session request information. For example, when HTTP sends a session request, identifying information can be informed as a query parameter to be transferred to an object on the basis of URI format.

[0114] If it is determined in step S92 that the access is from a dedicated client module, the flow branches to step S94; if not, the flow branches to step S93.

[0115] In step S93, whether the universal client extension module is to be downloaded to the universal client application is checked.

[0116] This function is realized by checking the right of the client that has requested the access by using the download objective client decision module 86.

[0117] If it is determined in step S93 that the module is not to be downloaded, the flow branches to step S94. If it is determined in step S93 that the client has the download right, the flow branches to step S95.

[0118] In step S94, in accordance with the type of requesting client the requested session is initialized by transferring the processing to an appropriate module from the interface supply module for dedicated client and the interface supply module for universal client, and session id for identifying the session is generated. After that, this session information is valid until a client sends a session termination request, and is managed by the server side on the basis of the session id. After this step, the flow advances to a termination process.

[0119] In step S95, the universal client extension module is downloaded to the requesting client, the session is once terminated to allow the client to install the module, and the client is caused to output a session request again.

[0120] Since this session is already open, in step S96 the session information is obtained from the session id, and the flow advances to step S97.

[0121] In step S97, on the basis of the session information the request for the designated image input device is transferred to and processed by the interface supply module for this client, and the result is returned to the requesting client via this module to terminate the processing.

[0122] Fig. 10 shows an example of the structure of management information of a session from a client to an image input device shared on the network.

[0123] Session id 101 is uniquely set to identify a desired one from a plurality of data.

[0124] Information 102 is used to identify an image input device which a client requests to access.

[0125] An area 103 stores information indicating the present status of the image input device as a result of setting information designated by sessions so far.

[0126] Identifying information 104 is used to identify the requesting client on the network.

[0127] Information 105 is used to identify the access right of the requesting client.

[0128] Time stamp information 106 is used to record the time the session is used last.

[0129] If the server device determines on the basis of this information that the recorded client has not accessed for a predetermined time or more from the last session, the server device deletes this session information by regarding this session as invalid. The purpose of this function is to prevent a specific client from occupying a shared image input device for long time periods without any access for long time.

[0130] An area 107 stores management information other than those described above.

[0131] Fig. 11 shows a display example when the client function extension module is incorporated into a universal client application.

[0132] This screen contains a universal client application display area 111 and an interface function 112 displayed by the client extension module and capable of designating an image input parameter.

[0133] A user can perform fine input settings for image input by designating desired control in this area by using a keyboard or a pointing device.

[0134] Fig. 12 shows a flow chart of internal processing when the client extension module is incorporated into a universal client application and the user sets image input parameters.

[0135] In step S121, whether the input from the user is designated to this extension module or the universal client application is checked.

[0136] If it is determined in step S121 that the input is designated to the extension module, the flow branches to step S122; if not, the flow advances to a termination process (hereinafter a termination process is not processing by the extension module but is transfer of processing to a universal application).

[0137] In step S122, whether the designated contents indicate that information needs to be transmitted to the server is checked. If YES in step S122, the flow branches to step S125; if NO in step S122, the flow branches to step S123.

[0138] In step S123, the input is processed only on the client side. That is, the client side performs some processing in accordance with the user's designation. Alternatively, the designated information is stored in an input information storage area set on the client side, so that the information can be transmitted to the server side at a later timing.

[0139] In step S124, whether this user input process is completed is checked. If it is determined in step S124 that the input state is continuing, the flow branches to step S124; if not, this process is completed, and the flow prepares for next input.

[0140] In step S125, the pieces of information stored in the input information storage area are obtained.

[0141] In step S126, the obtained pieces of information are collectively transmitted to the server.

[0142] In step S127, information obtained from the server as a result of step S126 is reflected on display in this module, and the processing is completed.

[0143] In the first embodiment as described above, it is possible to share image input devices on a network and control these image input devices from the client side by using a universal image in-out interface. This allows a user to be unconscious of whether a desired image input device exists on the network or is directly locally connected.

[0144] Consequently, the user on the client device side can use an image input device shared on the network without interrupting the work that he or she is currently performing.

[0145] Also, on the basis of information of an image input device existing on the network, virtual image input device connection information corresponding to a universal image in-out interface is automatically generated in a client device. Accordingly, the user on the client side can add a virtual device without installing individual connection information.

[0146] The load on the network can be reduced by properly using virtual image input device connection information in image reading sessions.

[0147] Furthermore, a desired one of a plurality of image input server devices existing in a network is selected, and virtual image input device connection information is generated only for the selected device. Since no wasteful input information is generated on the client device side, it is possible by providing this function to increase the processing efficiency and save the occupied area of the information.

[0148] In the second embodiment, an image input device is shared on a network, and a dedicated client module for supplying a universal image in-out interface to the image input device exists on the client device side, or a universal client application corresponding to a universal network protocol exists on the client device side. In either case, the user on the client side can use the image input device on the network by using one of these modules without interrupting the work that he or she is currently performing.

[0149] Additionally, the user on the client side can effectively use an image input device on the network by downloading an image reading designation extension function from the server device.

[0150] Furthermore, when the client side sends a download request for the module, whether the download is to be permitted is checked. With this function it is possible to set the security protection function.

[0151] The above embodiments can be applied to a case where programs are supplied to a system or an apparatus. In this case, the objects of the present invention are achieved by reading out and executing program codes stored in a storage medium to achieve the present invention by a computer (CPU or MPU) of the system or the apparatus.

[0152] The present invention also includes a case where a module provided by an OS (Operating System) running on the computer executes part of the processing when the computer executes the readout program codes.

[0153] The present invention is not limited to the above embodiments, and various modifications of the invention can be made.

[0154] The above programs may be transmitted over a communications channel as a signal carrying processor implementable instructions. A further aspect of the present invention therefore is constituted by such a signal.

Claims

1. An information processing apparatus connected to a network, comprising:

   searching means for searching for an image input server device on said network;
   obtaining means for obtaining, from an image input server device found by said searching means,

   information concerning said image input server device and information concerning an image input device locally connected to said image input server device;

   managing means for managing the information obtained by said obtaining means;
   generating means for generating virtual device information based on a universal image in-out interface from the information managed by said managing means;
   converting means for converting information of communication with said image input server device into said universal image in-out interface; and
   client means for executing an image input session for a desired image input device by using the virtual device information and the communication information.

2. An apparatus according to claim 1, wherein in the image input session said client means obtains static device control information from the virtual device information.

3. An apparatus according to claim 1, wherein a desired one of image input server devices found by said searching means is selected, and the virtual device information is generated only for the selected image input server device.

4. An apparatus according to claim 1, wherein said universal image in-out interface is TWAIN.

5. An apparatus according to claim 4, wherein when generating a name of a TWAIN device, said generating means generates a name containing ID of an image input device and ID of an image input server device to which said image input device is connected.

6. A system comprising an image input server device which can be shared on a network and a network client device capable of controlling an image input device shared on said network,

   said image input server device comprising:

input device control means for controlling operation of an image input device connected to said network, and
network server means capable of converting information of said input device control means into a universal network protocol and providing the protocol to said network; and
said client device comprising:
client means corresponding to the universal network protocol, and
interface converting means for converting device control information, obtained from said server by using said client means, into a universal image in-out interface.

7. A system in which a client device and a plurality of image input server devices are connected via a network,

said client device comprising:
searching means for searching for an image input server device on said network,
obtaining means for obtaining, from an image input server device found by said searching means,

information conceming said image input server device and information concerning an image input device locally connected to said image input server device,

managing means for managing the information obtained by said obtaining means,
generating means for generating virtual device information based on a universal image in-out interface from the information managed by said managing means,
converting means for converting information of communication with said image input server device into said universal image in-out interface, and
client means for executing an image input session for a desired image input device by using the virtual device information and the communication information; and
said image input server device comprising:
connecting means for locally connecting an image input device, and
server means for controlling an image input device connected to said connecting means on the basis of information of communication with said client device.

8. A system according to claim 7, wherein in the image input session said client means obtains static device control information from the virtual device information.

9. A system according to claim 7, wherein a desired one of image input server devices found by said searching means is selected, and the virtual device information is generated only for the selected image input server device.

10. A system according to claim 7, wherein said universal image in-out interface is TWAIN.

11. A system according to claim 10, wherein when generating a name of a TWAIN device, said generating means generates a name containing ID of an image input device and ID of an image input server device to which said image input device is connected.

12. An information processing apparatus connected to a network, comprising:

connecting means for locally connecting an image input device;
control means for controlling said image input device;
server means for executing an image reading session in collaboration with said control means in response to a session request from a client device on said network; and
determining means for determining the type of requesting client module from the session request,

wherein said server means executes an image reading session based on the determination result from said determining means.

13. An apparatus according to claim 12, wherein when said determining means determines that said requesting client module is a universal client module, said server means downloads, to said requesting client device, an extension module for extending a function of said universal client module.

14. An apparatus according to claim 13, wherein said server means determines whether said extension module is to be downloaded by checking whether said requesting client device has said extension module.

15. An apparatus according to claim 13, wherein said server means determines whether said extension module is to be downloaded on the basis of access right of said requesting client device.

16. A system in which a plurality of client devices and a server device are connected via a network,

said client device comprising:
client means for sending an image reading session request to said image input server device,

and
converting means for converting information of communication with said image input server device into a universal image in-out interface; and
said image input server device comprising:

connecting means for locally connecting an image input device,
control means for controlling said image input device,
server means for executing an image reading session in collaboration with said control means in response to a session request from a client device on said network, and
determining means for determining the type of requesting client module from the session request,

wherein said server means executes an image reading session based on the determination result from said determining means.

17. A system according to claim 16, wherein when said determining means determines that said requesting client module is a universal client module, said server means downloads, to said requesting client device, an extension module for extending a function of said universal client module.

18. A system according to claim 17, wherein said server means determines whether said extension module is to be downloaded by checking whether said requesting client device has said extension module.

19. A system according to claim 17, wherein said server means determines whether said extension module is to be downloaded on the basis of access right of said requesting client device.

20. A system according to claim 17, wherein said universal interface is a TWAIN interface.

21. A method of using an image input device locally connected to an image input server device on a network, comprising the steps of:

searching for an image input server device on said network;
obtaining, from said image input server device found, information concerning said image input server device and information concerning an image input device locally connected to said image input server device;
holding the obtained information as management information;
generating virtual device information based on a universal image in-out interface from the management information;
converting information of communication with said image input server device into said universal image in-out interface; and
executing an image input session for a desired image input device by using the virtual device information and the communication information.

22. A method according to claim 21, wherein in the image input session, static device control information is obtained from the virtual device information.

23. A method according to claim 21, wherein a desired one of image input server devices found by said searching means is selected, and the virtual device information is generated only for the selected image input server device.

24. A method according to claim 21, wherein said universal image in-out interface is TWAIN.

25. A method according to claim 24, wherein when a name of a TWAIN device is to be generated as the virtual device information, a name containing ID of an image input device and ID of an image input server device to which said image input device is connected is generated.

26. A method of using an image input device locally connected to an image input server device on a network, comprising the steps of:

in a client device on said network,
searching for an image input server device on said network,
obtaining, from said image input server device found, information concerning said image input server device and information concerning an image input device locally connected to said image input server device,
holding the obtained information as management information,
generating virtual device information based on a universal image in-out interface from the management information,
converting information of communication with said image input server device into said universal image in-out interface, and
executing an image input session for a desired image input device by using the virtual device information and the communication information; and
in said image input server device,
controlling said image input device on the basis of information of communication with said client device.

27. A method according to claim 26, wherein in the image input session, static device control information is obtained from the virtual device information.

28. A method according to claim 26, wherein a desired one of image input server devices found by said searching means is selected, and the virtual device information is generated only for the selected image input server device.

29. A method according to claim 26, wherein said universal image in-out interface is TWAIN.

30. A method according to claim 29, wherein when a name of a TWAIN device is to be generated as the virtual device information, a name containing ID of an image input device and ID of an image input server device to which said image input device is connected is generated.

31. A method of sharing a locally connected image input device on a network, comprising the steps of:

receiving a session request from a client device on said network;
determining the type of requesting client module from the received session request; and
executing an image reading session on the basis of the determination result by controlling said image input device.

32. A method according to claim 31, wherein when in the image reading session the determination result indicates that said requesting client module is a universal client module, an extension module for extending a function of said universal client module is downloaded to said requesting client device.

33. A method according to claim 32, wherein whether said extension module is to be downloaded is determined by checking whether said requesting client device has said extension module.

34. A method according to claim 32, wherein whether said extension module is to be downloaded is determined on the basis of access right of said requesting client device.

35. A method according to claim 32, wherein said universal image in-out interface is TWAIN.

36. A method of using an image input device locally connected to an image input server on a network, comprising the steps of:

in a client device on said network,
sending an image reading session request to said image input server device, and
converting information of communication with said image input server device into a universal image in-out interface; and
in said image input server device,
receiving the session request from said client device,
determining the type of requesting client module from the received session request, and
executing an image reading session on the basis of the determination result by controlling said image input device.

37. A method according to claim 36, wherein when in the image reading session the determination result indicates that said requesting client module is a universal client module, an extension module for extending a function of said universal client module is downloaded to said requesting client device.

38. A method according to claim 36, wherein whether said extension module is to be downloaded is determined by checking whether said requesting client device has said extension module.

39. A method according to claim 36, wherein whether said extension module is to be downloaded is determined on the basis of access right of said requesting client device.

40. A method according to claim 36, wherein said universal image in-out interface is a TWAIN interface.

41. A storage medium storing a computer program executed by a computer of an information processing apparatus connected to a network, said computer program comprising the steps of:

a process of searching for an image input server device on said network;
a process of obtaining, from said image input server device found, information concerning said image input server device and information concerning an image input device locally connected to said image input server device;
a process of holding the obtained information as management information;
a process of generating virtual device information based on a universal image in-out interface from the management information;
a process of converting information of communication with said image input server device into said universal image in-out interface; and
a process of executing an image input session for a desired image input device by using the virtual device information and the communication information.

42. A medium according to claim 41, wherein in the image input session, static device control information is obtained from the virtual device information.

43. A medium according to claim 41, wherein a desired one of said found image input server devices is selected, and the virtual device information is generated only for the selected image input server device.

44. A medium according to claim 41, wherein said universal image in-out interface is TWAIN.

45. A medium according to claim 44, wherein when a name of a TWAIN device is to be generated as the virtual device information, a name containing ID of an image input device and ID of an image input server device to which said image input device is connected is generated.

46. A method of sharing a locally connected image input device on a network, comprising the steps of:

a process of receiving a session request from a client device on said network;
a process of determining the type of requesting client module from the received session request; and
a process of executing an image reading session on the basis of the determination result by using said image input device.

47. A method according to claim 46, wherein when in the image reading session the determination result indicates that said requesting client module is a universal client module, an extension module for extending a function of said universal client module is downloaded to said requesting client device.

48. A method according to claim 47, wherein whether said extension module is to be downloaded is determined by checking whether said requesting client device has said extension module.

49. A method according to claim 47, wherein whether said extension module is to be downloaded is determined on the basis of access right of said requesting client device.

50. A method according to claim 46, wherein said universal image in-out interface is TWAIN.

51. An information processing apparatus connected to a network, comprising:

searching means for searching for an image input server device on said network;
obtaining means for obtaining, from an image input server device found by said searching means, information concerning an image input device connected to said image input server device;

generating means for generating virtual device information based on a universal image in-out interface from the information obtained by said obtaining means;
converting means for converting information of communication with said image input server device into said universal image in-out interface; and
client means for executing an image input session for a desired image input device by using the virtual device information and the communication information.

52. A method of operating an image input device remotely via a network connection by using virtual device information derived from device information obtained from a server connected locally to the image input device.

53. A signal carrying processor implementable instructions for carrying out a method as claimed in any of claims 21 to 40, 45 to 50, and 52.

FIG.1

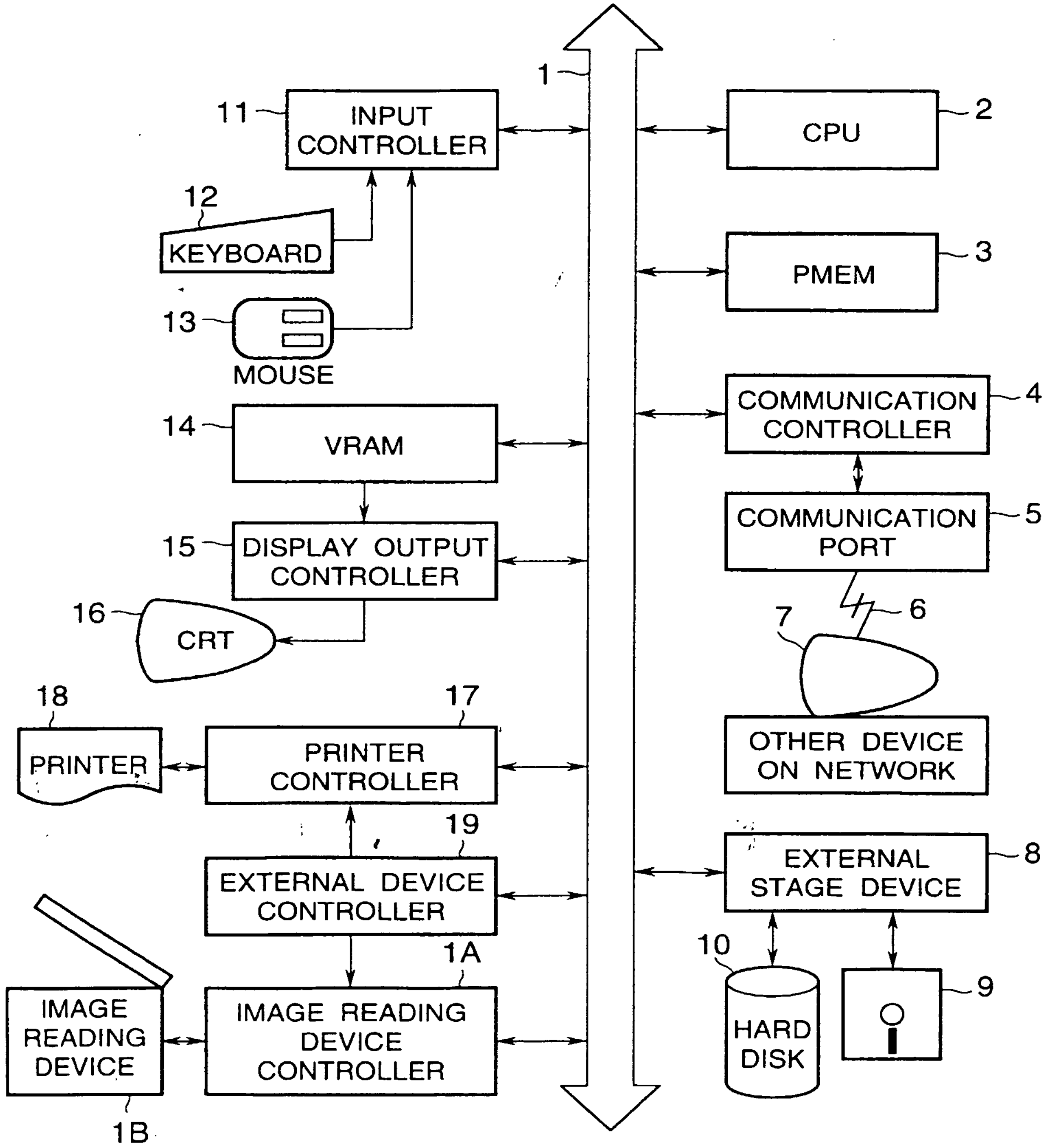

1
INPUT CONTROLLER
11
CPU
2
12
KEYBOARD
PMEM
3
13
MOUSE
COMMUNICATION CONTROLLER
4
14
VRAM
COMMUNICATION PORT
5
15
DISPLAY OUTPUT CONTROLLER
6
7
16
CRT
18
17
PRINTER
PRINTER CONTROLLER
OTHER DEVICE ON NETWORK
19
EXTERNAL DEVICE CONTROLLER
EXTERNAL STAGE DEVICE
8
1A
10
IMAGE READING DEVICE
IMAGE READING DEVICE CONTROLLER
HARD DISK
9
1B

FIG.2

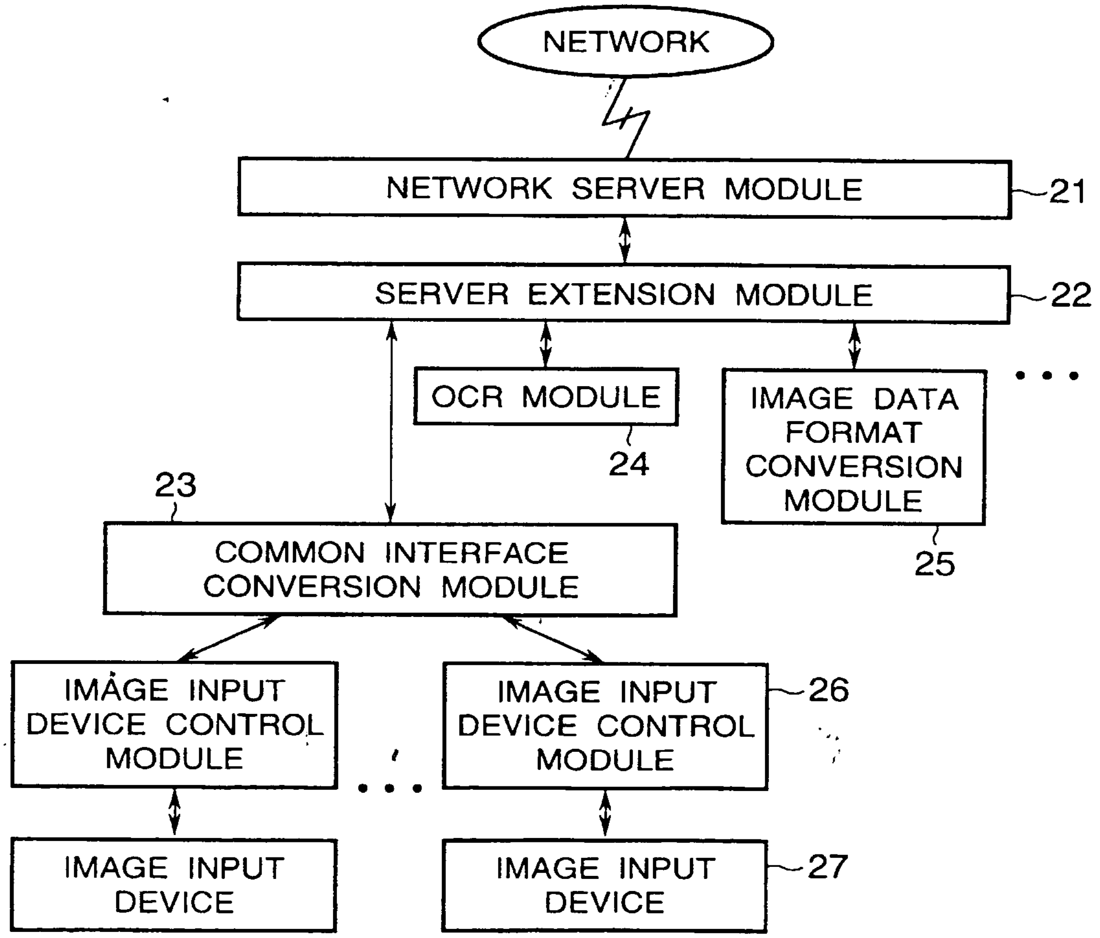

NETWORK
NETWORK SERVER MODULE
21
SERVER EXTENSION MODULE
22
OCR MODULE
24
IMAGE DATA FORMAT CONVERSION MODULE
25
23
COMMON INTERFACE CONVERSION MODULE
IMAGE INPUT DEVICE CONTROL MODULE
IMAGE INPUT DEVICE CONTROL MODULE
26
IMAGE INPUT DEVICE
IMAGE INPUT DEVICE
27

FIG.3

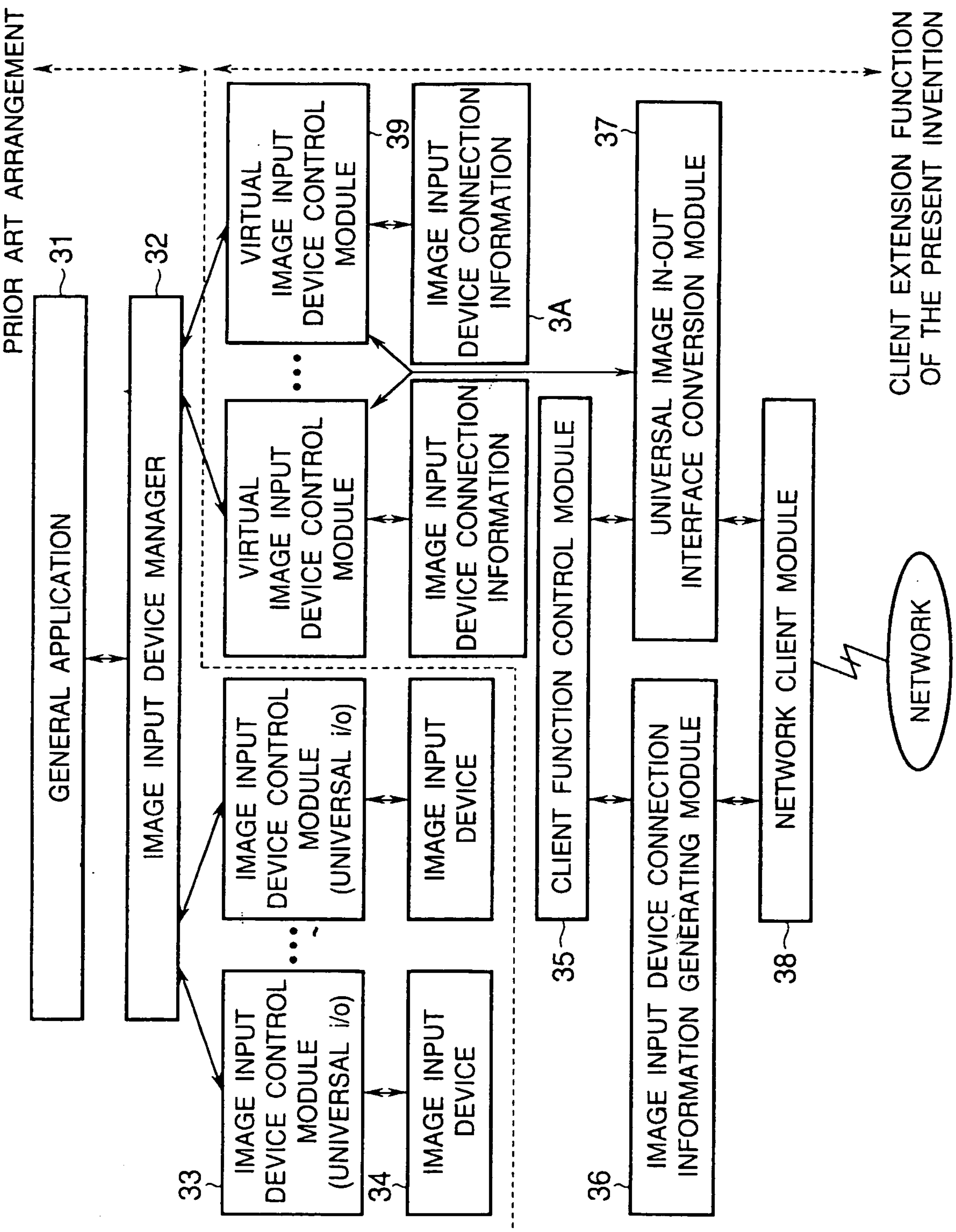

PRIOR ART ARRANGEMENT
CLIENT EXTENSION FUNCTION OF THE PRESENT INVENTION
31
GENERAL APPLICATION
32
IMAGE INPUT DEVICE MANAGER
33
IMAGE INPUT DEVICE CONTROL MODULE (UNIVERSAL i/o)
IMAGE INPUT DEVICE CONTROL MODULE (UNIVERSAL i/o)
34
IMAGE INPUT DEVICE
IMAGE INPUT DEVICE
VIRTUAL IMAGE INPUT DEVICE CONTROL MODULE
VIRTUAL IMAGE INPUT DEVICE CONTROL MODULE
39
IMAGE INPUT DEVICE CONNECTION INFORMATION
IMAGE INPUT DEVICE CONNECTION INFORMATION
3A
35
CLIENT FUNCTION CONTROL MODULE
36
IMAGE INPUT DEVICE CONNECTION INFORMATION GENERATING MODULE
37
UNIVERSAL IMAGE IN-OUT INTERFACE CONVERSION MODULE
38
NETWORK CLIENT MODULE
NETWORK

FIG.4

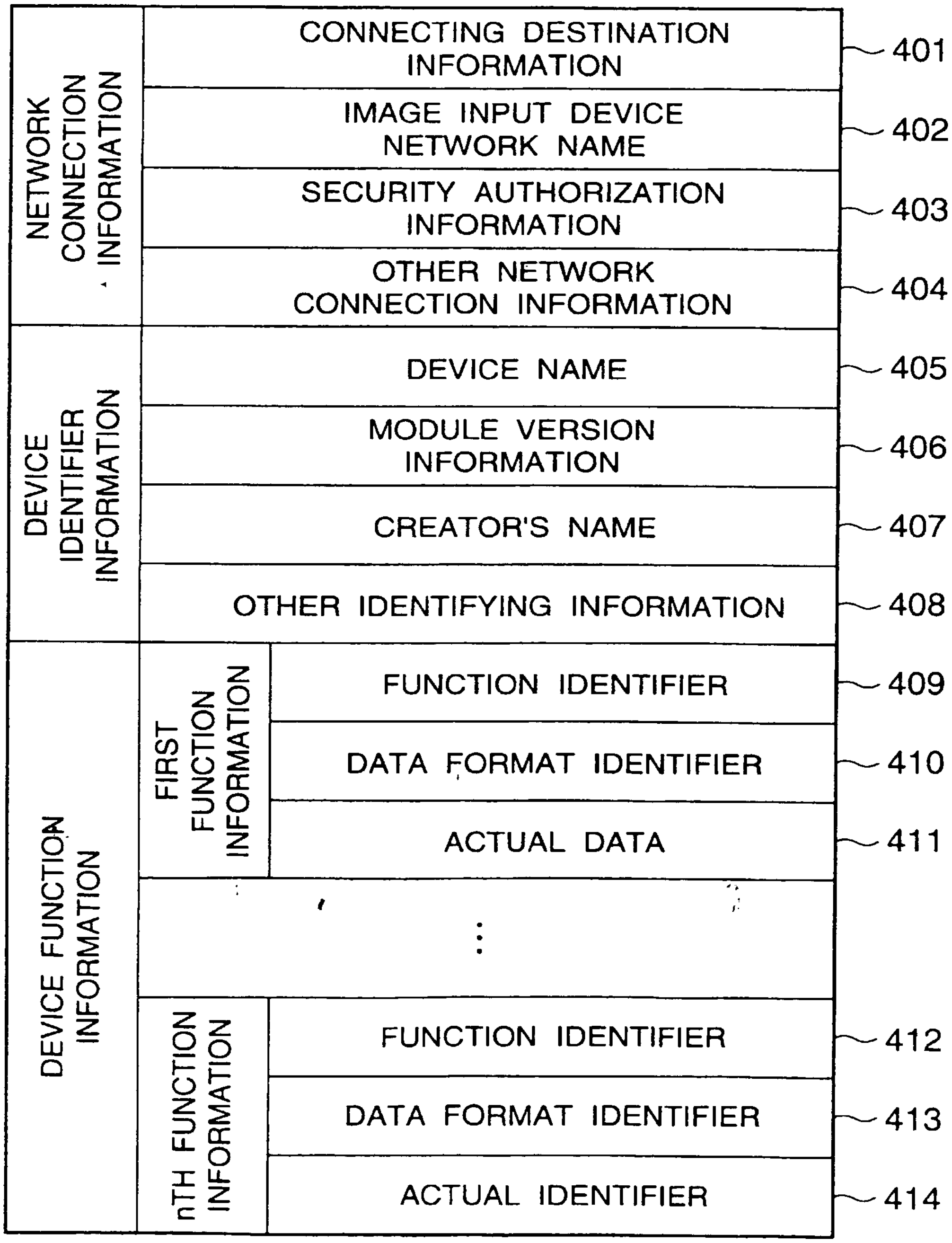

NETWORK CONNECTION INFORMATION
CONNECTING DESTINATION INFORMATION
401
IMAGE INPUT DEVICE NETWORK NAME
402
SECURITY AUTHORIZATION INFORMATION
403
OTHER NETWORK CONNECTION INFORMATION
404
DEVICE IDENTIFIER INFORMATION
DEVICE NAME
405
MODULE VERSION INFORMATION
406
CREATOR'S NAME
407
OTHER IDENTIFYING INFORMATION
408
DEVICE FUNCTION INFORMATION
FIRST FUNCTION INFORMATION
FUNCTION IDENTIFIER
409
DATA FORMAT IDENTIFIER
410
ACTUAL DATA
411
⋮
nTH FUNCTION INFORMATION
FUNCTION IDENTIFIER
412
DATA FORMAT IDENTIFIER
413
ACTUAL IDENTIFIER
414

FIG.5

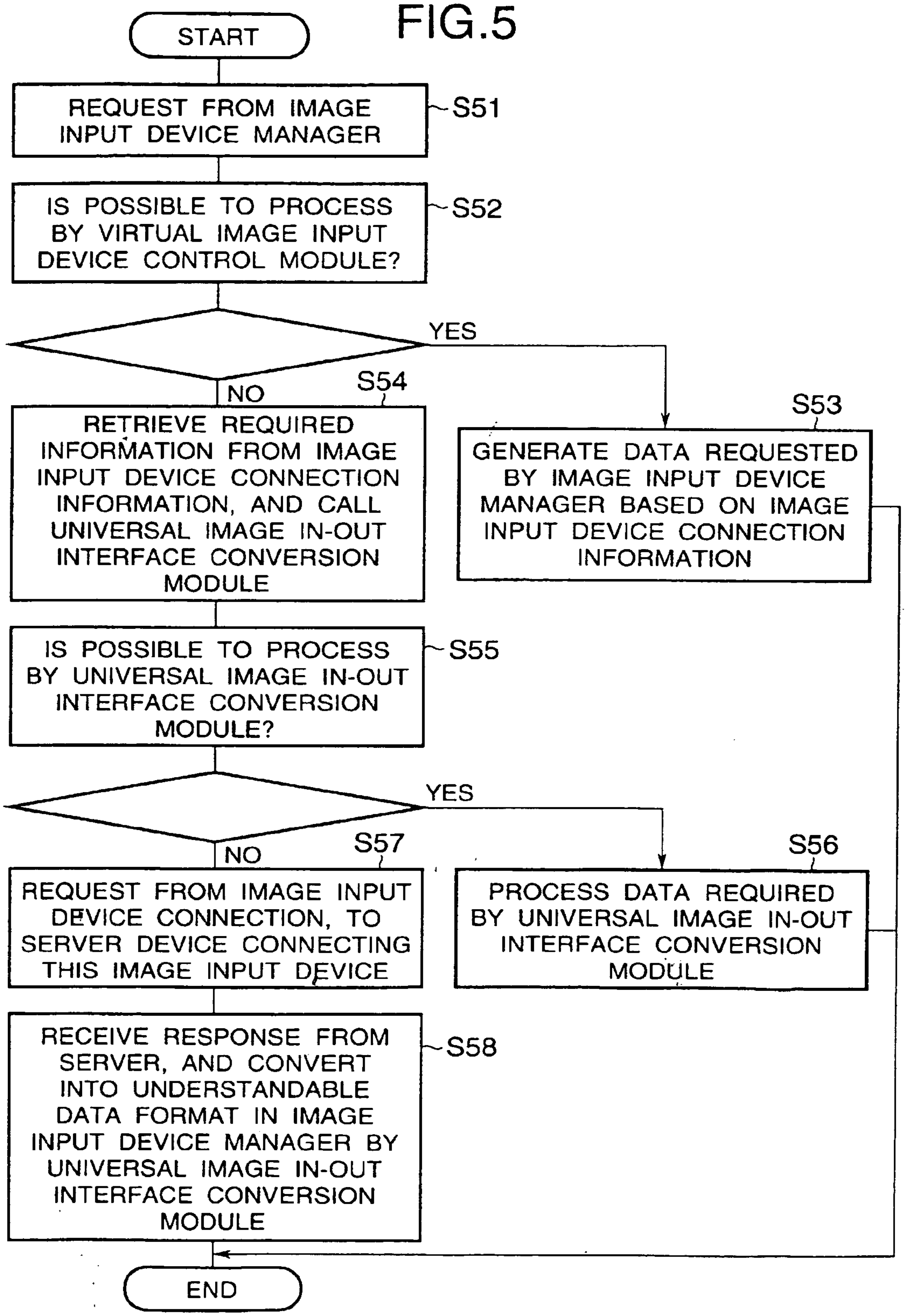

START
REQUEST FROM IMAGE INPUT DEVICE MANAGER
S51
IS POSSIBLE TO PROCESS BY VIRTUAL IMAGE INPUT DEVICE CONTROL MODULE?
S52
YES
NO
S54
RETRIEVE REQUIRED INFORMATION FROM IMAGE INPUT DEVICE CONNECTION INFORMATION, AND CALL UNIVERSAL IMAGE IN-OUT INTERFACE CONVERSION MODULE
S53
GENERATE DATA REQUESTED BY IMAGE INPUT DEVICE MANAGER BASED ON IMAGE INPUT DEVICE CONNECTION INFORMATION
IS POSSIBLE TO PROCESS BY UNIVERSAL IMAGE IN-OUT INTERFACE CONVERSION MODULE?
S55
YES
NO
S57
REQUEST FROM IMAGE INPUT DEVICE CONNECTION, TO SERVER DEVICE CONNECTING THIS IMAGE INPUT DEVICE
S56
PROCESS DATA REQUIRED BY UNIVERSAL IMAGE IN-OUT INTERFACE CONVERSION MODULE
RECEIVE RESPONSE FROM SERVER, AND CONVERT INTO UNDERSTANDABLE DATA FORMAT IN IMAGE INPUT DEVICE MANAGER BY UNIVERSAL IMAGE IN-OUT INTERFACE CONVERSION MODULE
S58
END

FIG.6

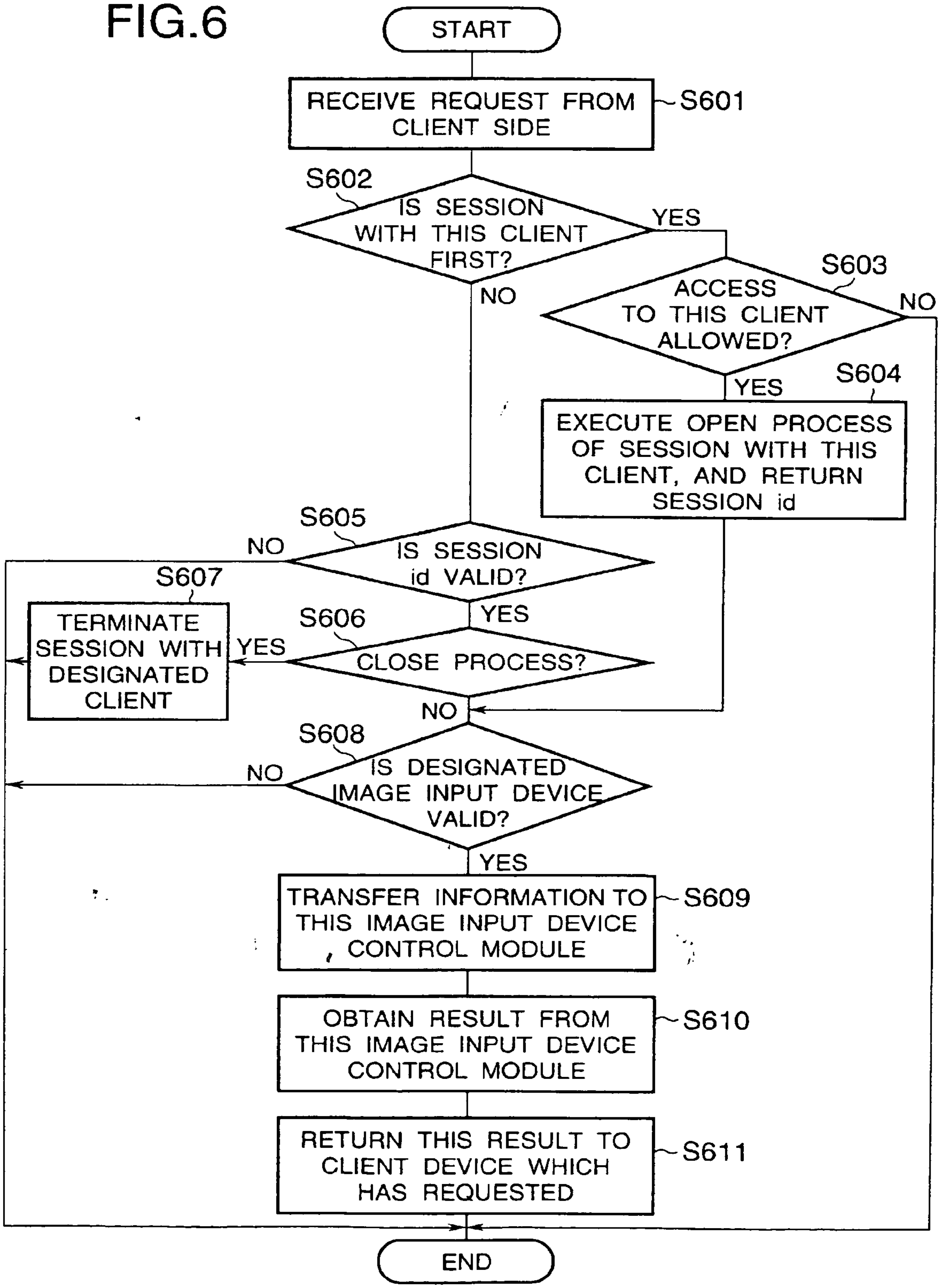
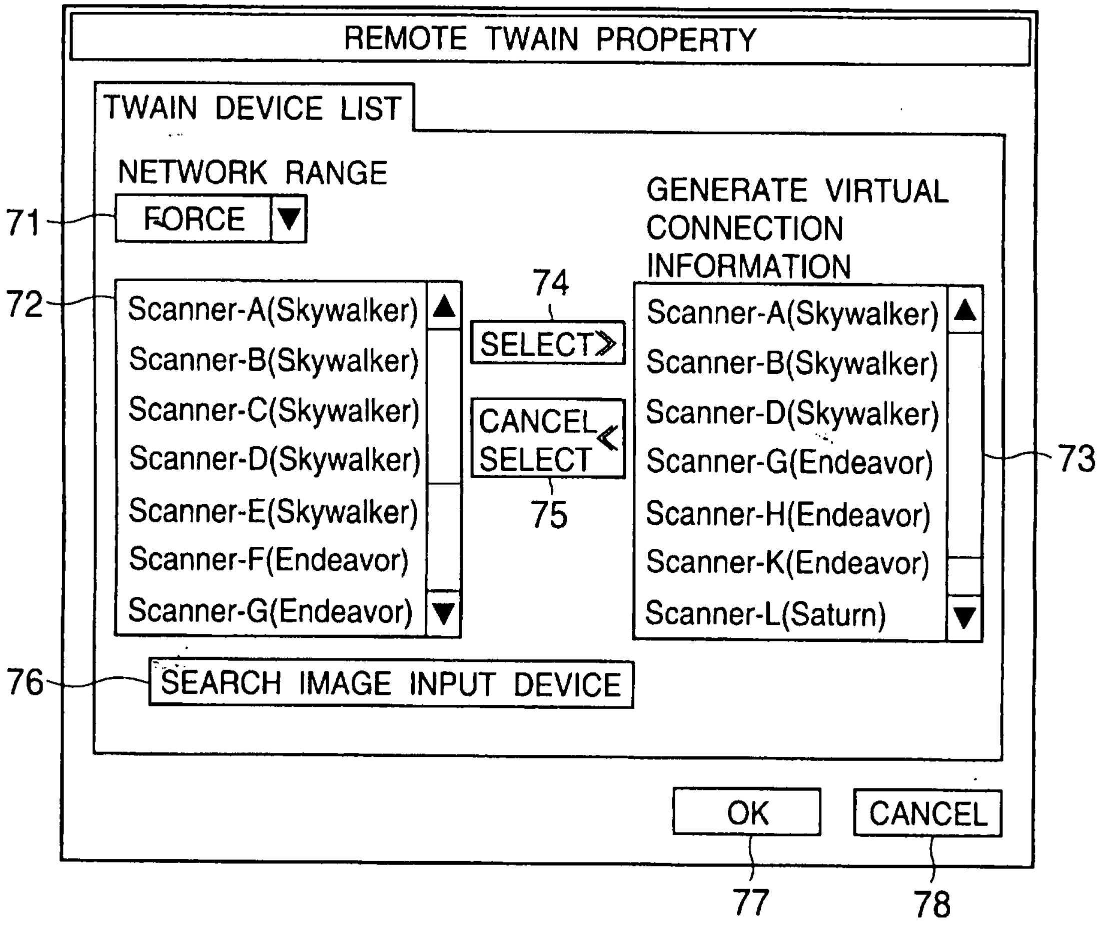

START
RECEIVE REQUEST FROM CLIENT SIDE
S601
S602
IS SESSION WITH THIS CLIENT FIRST?
YES
NO
S603
ACCESS TO THIS CLIENT ALLOWED?
NO
YES
S604
EXECUTE OPEN PROCESS OF SESSION WITH THIS CLIENT, AND RETURN SESSION id
S605
NO
IS SESSION id VALID?
YES
S607
TERMINATE SESSION WITH DESIGNATED CLIENT
S606
YES
CLOSE PROCESS?
NO
S608
NO
IS DESIGNATED IMAGE INPUT DEVICE VALID?
YES
TRANSFER INFORMATION TO THIS IMAGE INPUT DEVICE CONTROL MODULE
S609
OBTAIN RESULT FROM THIS IMAGE INPUT DEVICE CONTROL MODULE
S610
RETURN THIS RESULT TO CLIENT DEVICE WHICH HAS REQUESTED
S611
END

FIG.7



REMOTE TWAIN PROPERTY

TWAIN DEVICE LIST

NETWORK RANGE

FORCE ▼

Scanner-A(Skywalker)
Scanner-B(Skywalker)
Scanner-C(Skywalker)
Scanner-D(Skywalker)
Scanner-E(Skywalker)
Scanner-F(Endeavor)
Scanner-G(Endeavor)

SELECT≫

CANCEL SELECT≪

GENERATE VIRTUAL CONNECTION INFORMATION

Scanner-A(Skywalker)
Scanner-B(Skywalker)
Scanner-D(Skywalker)
Scanner-G(Endeavor)
Scanner-H(Endeavor)
Scanner-K(Endeavor)
Scanner-L(Saturn)

SEARCH IMAGE INPUT DEVICE

OK

CANCEL

FIG.8

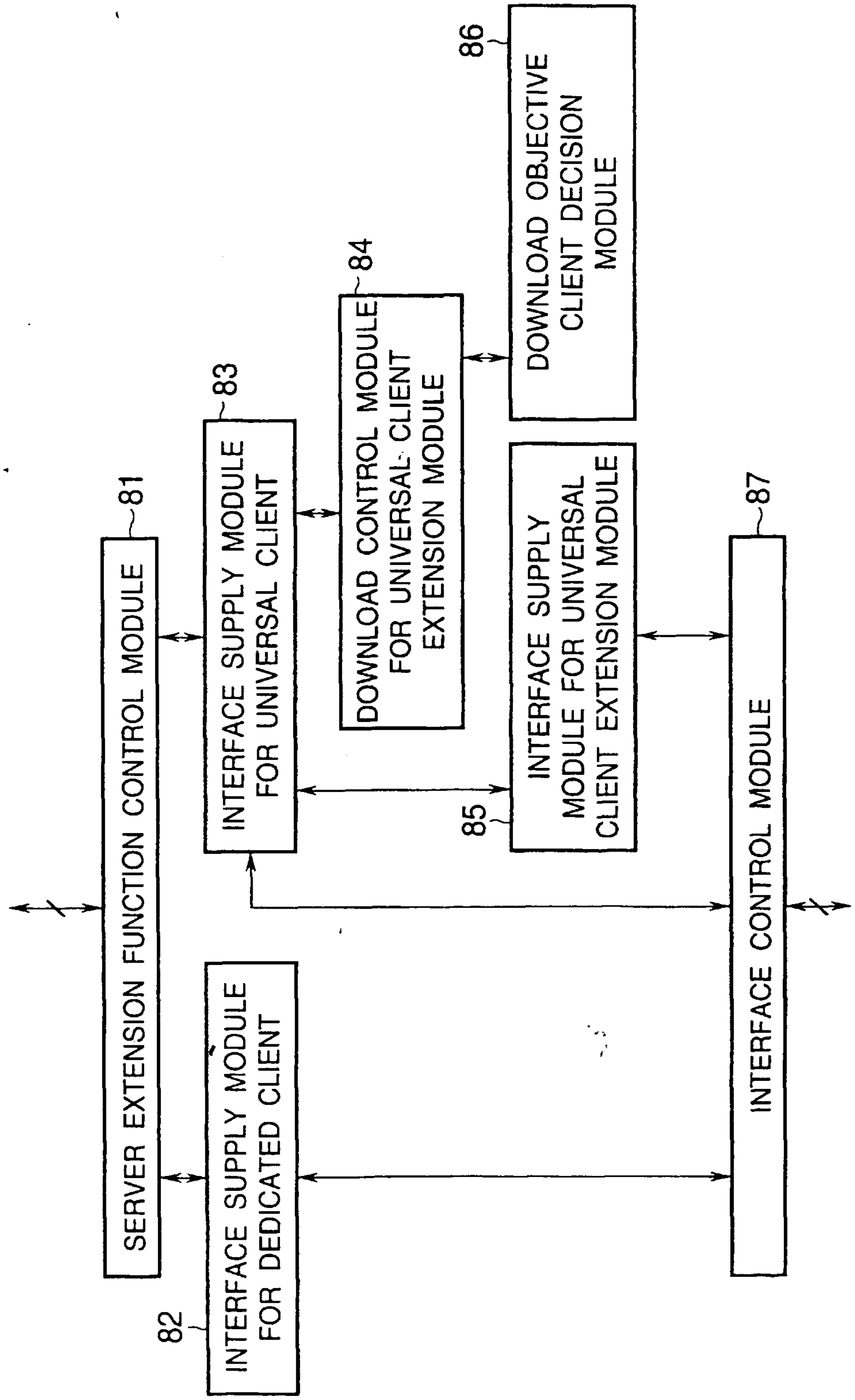

SERVER EXTENSION FUNCTION CONTROL MODULE
81
INTERFACE SUPPLY MODULE FOR DEDICATED CLIENT
82
INTERFACE SUPPLY MODULE FOR UNIVERSAL CLIENT
83
DOWNLOAD CONTROL MODULE FOR UNIVERSAL CLIENT EXTENSION MODULE
84
INTERFACE SUPPLY MODULE FOR UNIVERSAL CLIENT EXTENSION MODULE
85
DOWNLOAD OBJECTIVE CLIENT DECISION MODULE
86
INTERFACE CONTROL MODULE
87

FIG.9

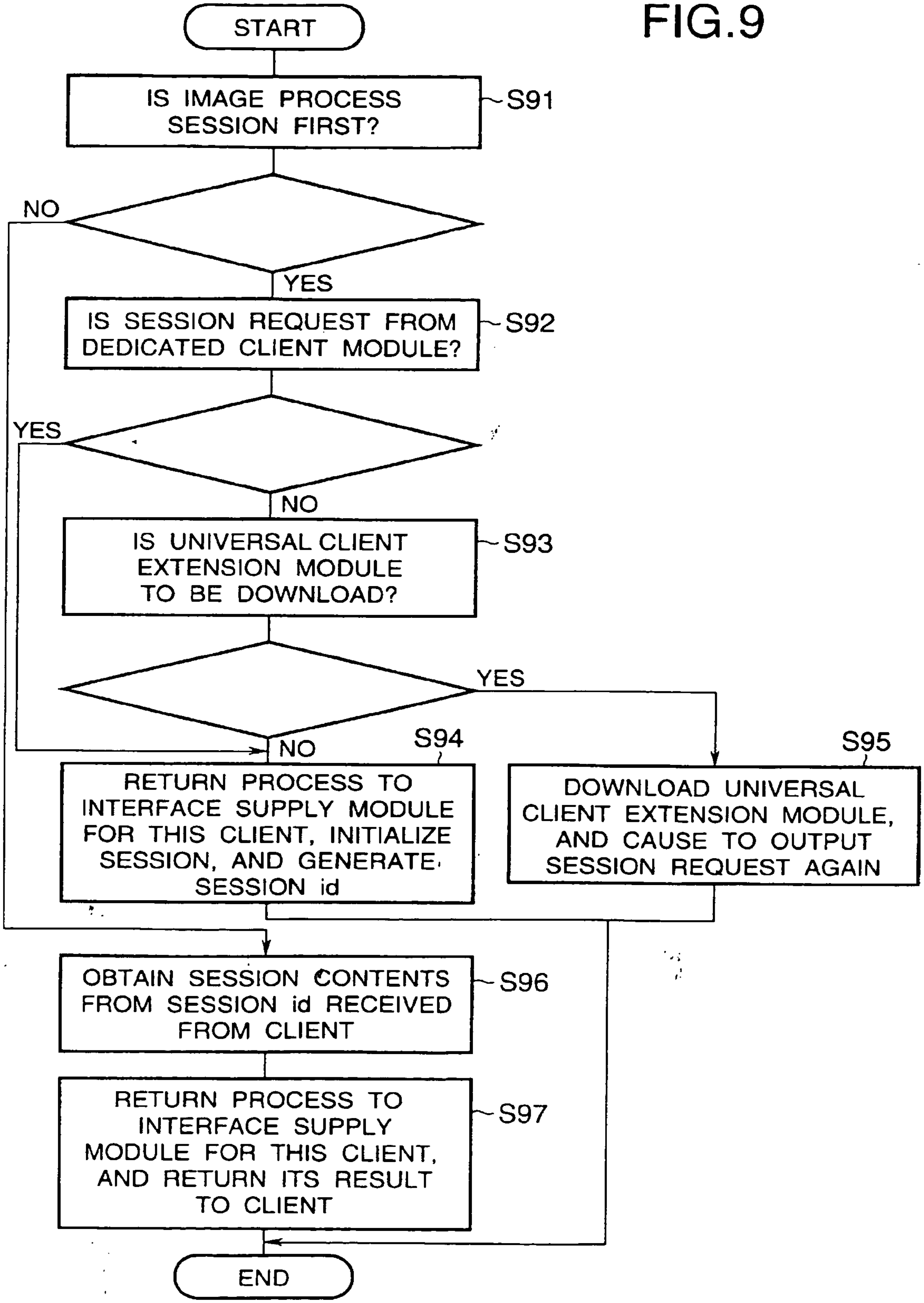

START
IS IMAGE PROCESS SESSION FIRST?
S91
NO
YES
IS SESSION REQUEST FROM DEDICATED CLIENT MODULE?
S92
YES
NO
IS UNIVERSAL CLIENT EXTENSION MODULE TO BE DOWNLOAD?
S93
YES
NO
S94
RETURN PROCESS TO INTERFACE SUPPLY MODULE FOR THIS CLIENT, INITIALIZE SESSION, AND GENERATE SESSION id
S95
DOWNLOAD UNIVERSAL CLIENT EXTENSION MODULE, AND CAUSE TO OUTPUT SESSION REQUEST AGAIN
OBTAIN SESSION CONTENTS FROM SESSION id RECEIVED FROM CLIENT
S96
RETURN PROCESS TO INTERFACE SUPPLY MODULE FOR THIS CLIENT, AND RETURN ITS RESULT TO CLIENT
S97
END

FIG.10

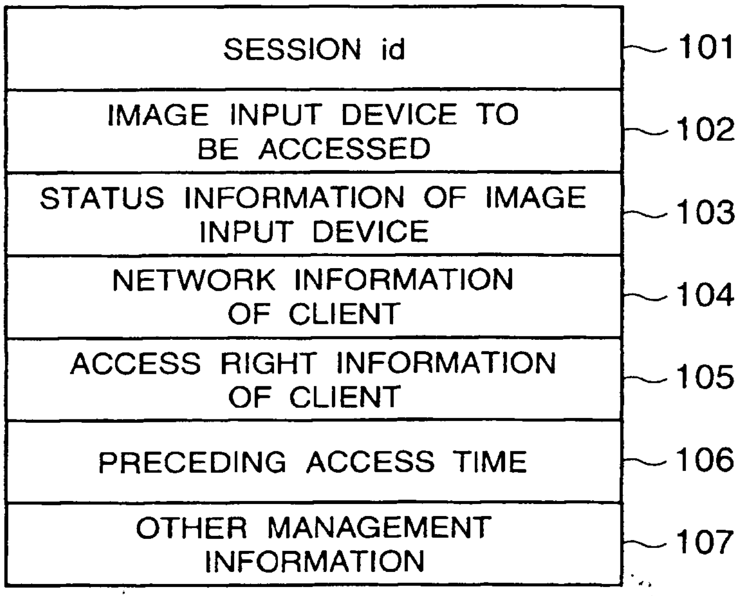
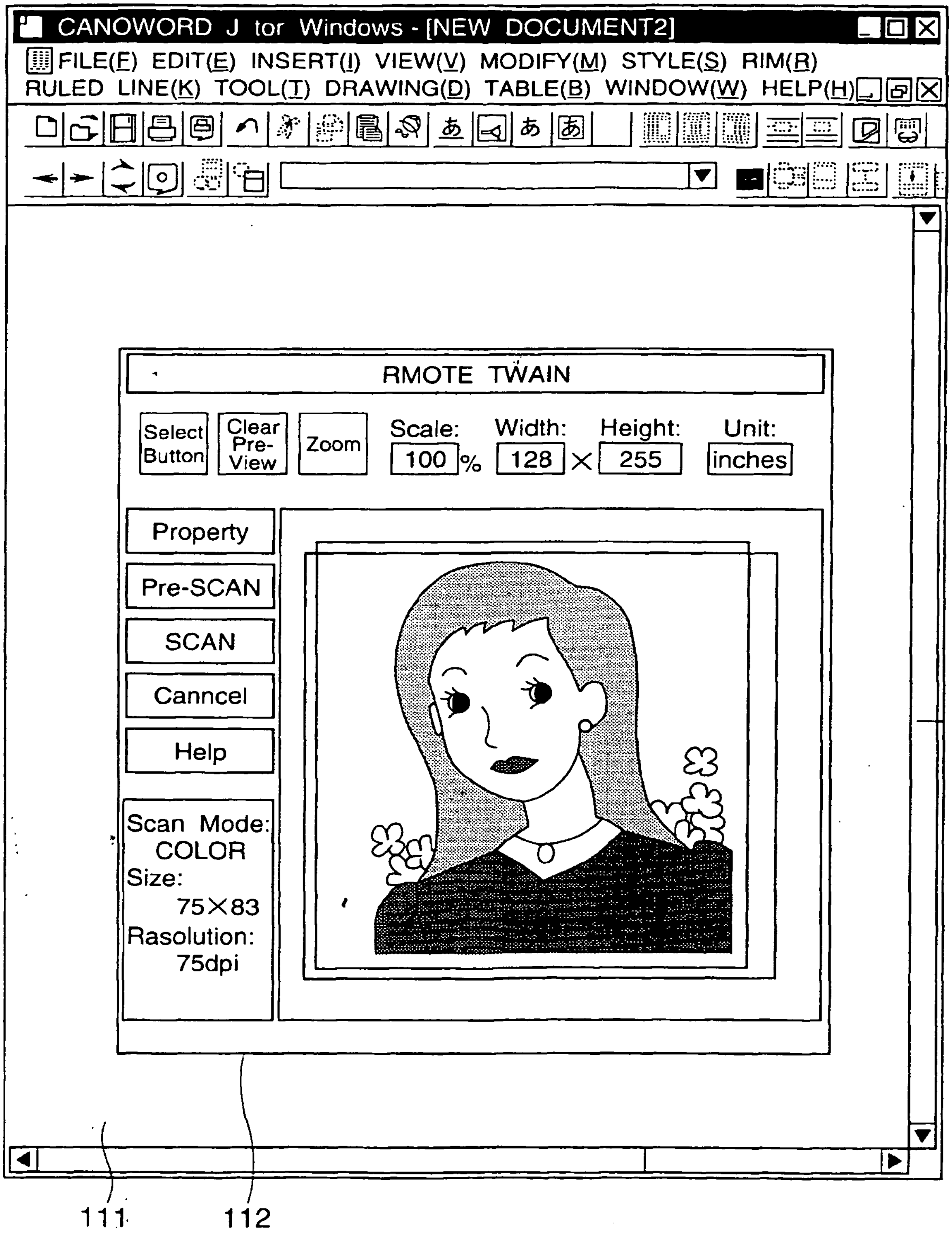

SESSION id
101
IMAGE INPUT DEVICE TO BE ACCESSED
102
STATUS INFORMATION OF IMAGE INPUT DEVICE
103
NETWORK INFORMATION OF CLIENT
104
ACCESS RIGHT INFORMATION OF CLIENT
105
PRECEDING ACCESS TIME
106
OTHER MANAGEMENT INFORMATION
107

FIG.11


CANOWORD J tor Windows - [NEW DOCUMENT2]
FILE(F) EDIT(E) INSERT(I) VIEW(V) MODIFY(M) STYLE(S) RIM(R)
RULED LINE(K) TOOL(T) DRAWING(D) TABLE(B) WINDOW(W) HELP(H)
あ
あ
あ
RMOTE TWAIN
Select Button
Clear Pre-View
Zoom
Property
Pre-SCAN
SCAN
Canncel
Help
111
112

Scale: 100 %

Width: 128 × Height: 255

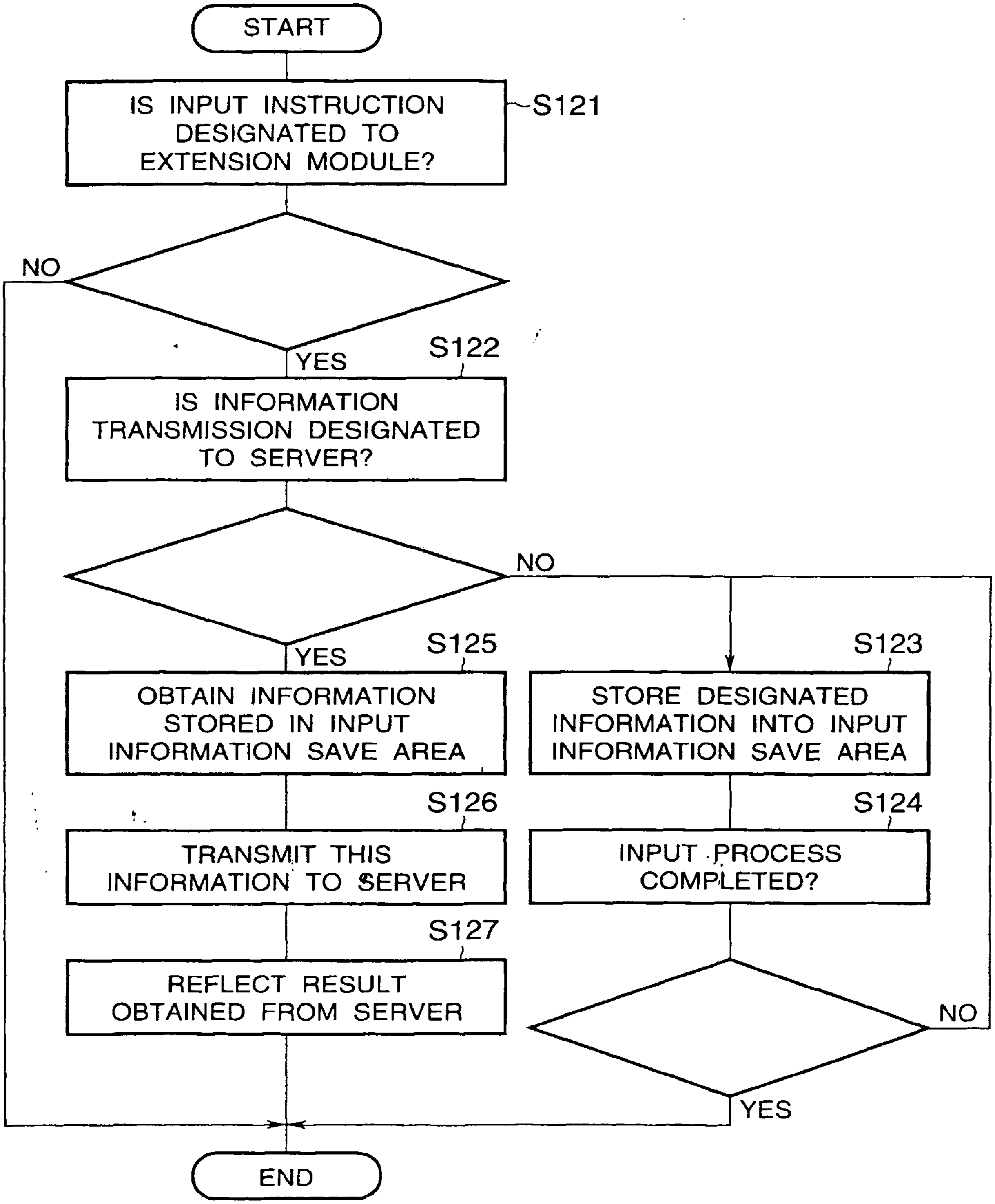
Unit: inches

Scan Mode: COLOR

Size: 75×83

Rasolution: 75dpi

FIG.12


START
IS INPUT INSTRUCTION DESIGNATED TO EXTENSION MODULE?
S121
NO
YES
S122
IS INFORMATION TRANSMISSION DESIGNATED TO SERVER?
NO
YES
S125
OBTAIN INFORMATION STORED IN INPUT INFORMATION SAVE AREA
S123
STORE DESIGNATED INFORMATION INTO INPUT INFORMATION SAVE AREA
S126
TRANSMIT THIS INFORMATION TO SERVER
S124
INPUT PROCESS COMPLETED?
S127
REFLECT RESULT OBTAINED FROM SERVER
NO
YES
END